

Заключение. Один из главных факторов, влияющих на эффективность нейронной сети является выделение каждого пикселя исследуемого объекта на изображении при подготовки данных для обучающей выборки, а также имеющиеся вычислительные мощности. Факт того, что изображения на обучающей выборки содержали исследуемый объект по центру изображения, не повлиял на способность обученной нейронной сети определять необработанные участки поля, располагающиеся по краям камеры (рисунок 3).

ЛИТЕРАТУРА

1. J. R. Rosell-Polo, F. A. Cheein, E. Gregorio, D. Andujar, L. Puigdomenech, J. Masip, and A. Escol ` a, "Advances in structured light ` sensors applications in precision agriculture and livestock farming," *Advances in Agronomy*, vol. 133, pp. 71–112, 2015.
2. R. Xiang, H. Jiang, and Y. Ying, "Recognition of clustered tomatoes based on binocular stereo vision," *Computers and Electronics in Agriculture*, vol. 106, pp. 75–90, 2014.
3. Геоскан. Сельское хозяйство. – URL: <https://www.geoscan.aero/ru/application/agriculture> (дата обращения 10.10.2018).
4. Mask R-CNN. – URL: <https://arxiv.org/abs/1703.06870> (дата обращения 10.10.2018).
5. Tensorflow Object Detection API. – URL: https://github.com/tensorflow/models/tree/master/research/object_detection (дата обращения 8.10.2018).

АЛГОРИТМ ОТСЛЕЖИВАНИЯ И ВЫДЕЛЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ В ВИДЕОПОТОКЕ

К.П. Лошаков

(г. Томск, Томский политехнический университет)

e-mail: kpl2@tpu.ru

ALGORITHM OF TRACKING AND DETECTION MOVING OBJECTS IN VIDEO STREAM

K.P. Loshakov

(Tomsk, Tomsk Polytechnik University)

Abstract. This paper describes the solution detection moving people in a videostream. Instruments of OpenCV library were used. As the result this approach shown high accuracy of people detection and no high images processing time.

Keywords: OpenCV, python, detection algorithm, frames comparison, real-time processing, BackgroundSubtractionMOG2, numpy, Mask R-CNN.

Введение. С увеличением количества камер на улицах, в переулках, торговых центрах и прочих местах растет возможность записывать все, что происходит в месте видеосъемки. Соответственно, появляется возможность применения технического зрения для решения ряда задач, основанных на анализе видеопотока. [1] Особенность внедрения технического зрения заключается в возможности анализа видеопотока. Например, решение задачи слежения за изменениями обстановки в кадре (появление объекта, его движение), что напрямую связано с возможностью обнаружения подозрительного поведения человека в кадре, находящегося магазине, банке, аэропорту или другом общественном месте. Также анализируя видеопоток, достигается предприятие каких-либо действий (отсылка информации о подозрительном поведении объекта в пункт охраны заведения). Другая целевая задача – определение скорости перемещения автомобилей по автомагистрали, для контроля соблюдения водителями правил скоростного движения.[2] Также техническое зрение используется для создания эффектив-

ных систем автоматизированного контроля технологических процессов с целью обеспечения заданных показателей точности и быстродействия.[3]

В целом, техническое зрение охватило огромное количество разнообразных сфер жизни человека от промышленности до сельского хозяйства и медицины.[4-6]

Данная работа предлагает алгоритм для слежения за изменяющими свое положение в кадре объектами, путем заключения их в прямоугольные области.

Известно, что для решения подобных задач применяются каскады Хаара. [7] Цель применения состоит в нахождении определенного объекта на изображении. В рамках представляемой работы выполняется сравнение разницы между кадрами и выделении именно объектов изменяющих положение через установленный промежуток времени. Время выполнения алгоритма программы составляет 0,058 секунд. Можно уменьшить время обработки, используя сочетание CPU + GPU с планировщиком (scheduler).[8] В качестве другого и возможно более быстродействующего алгоритма для распознавания движений в кадре может быть применен метод k-ближайших соседей. Метод в данной статье может быть не всегда пригоден в случае динамического заднего плана, но конкретных исследований не проводилось.[9] В сравнении с методом использования каскада Хаара не нужно подготавливать примеры позитивных и негативных изображений, то есть не нужно использовать определенные инструменты, при использовании которых на выходе создаются XML файлы, которые затем обрабатываются командами классификатора каскадов.[10–11]

Описание алгоритма нахождения координат прямоугольников, обрамляющих объекты. Предлагаемый в рамках работы алгоритм слежения предназначен для обнаружения людей в помещении. Он является универсальным и его можно использовать для определения людей или других движущихся объектов в различных местах съемки.

Обобщенная блок-схема алгоритма программы движущихся объектов представлена на рис. 1.



Рис. 1. Обобщенная блок-схема алгоритма программы нахождения движущихся объектов

Основные этапы алгоритма, блок-схема которого показана на рис.1, следующие:

- вычитание заднего фона, с помощью применения алгоритма BackgroundSubtractorMOG2 библиотеки OpenCV к двум кадрам; [12 – 14]
- устранение шумов с использованием морфологической трансформации morphologyEx библиотеки OpenCV с параметром MORPH_OPEN;
- определение кривых на изображении, вдоль которых происходит резкое изменение яркости или других видов неоднородностей с помощью алгоритма определения границ Canny библиотеки OpenCV.

Написание кода программы для разбиения матрицы пикселей кадра на зоны вдоль горизонта, в которых находятся границы объектов. Поиск в зонах границ с учетом того, что в зоне может находиться несколько объектов.

Для эффективных численных вычислений при работе с матрицей пикселей изображения используется библиотека Numpy.[15] Сначала создается матрица-строка, заполненная нулями, которая впоследствии проходит вдоль изображения по горизонтали, находя и записывая в новый numpy массив индексы столбцов, в которых какое-либо значение не равно нулю. Если размер нового массива не равен нулю, то начинается цикл, который выполняется пока значение инкрементирующийся переменной не достигнет величины длины размерности

массива индексов. В данном цикле осуществляется запись граничных индексов объектов по горизонтали. Величина дистанции между объектами может быть задана пользователем. После обнаружения границ всех объектов или группы объектов по горизонтали происходит выделение сегмента кадра, в котором находится объект. Далее происходят операции аналогичные тем, что проводились при нахождении границ по горизонтали. В конечном счете, происходит запись в лист координат точек крайнего левого верхнего и крайнего правого нижнего углов объекта или группы объектов. После нахождения всех объектов алгоритм возвращает список координат (4-х точек) всех обрамляющих объекты прямоугольников.

Экспериментальное исследование разработанного алгоритма. Для исследования эффективности предлагаемого алгоритма было обработано видео длительностью 150 секунд с шириной кадра 1280 пикселей и высотой кадра 720 пикселей.

Видео выбрано высокой четкости (HD) для оценивания быстродействия алгоритма при обработке видео с популярным форматом.

На рис. 2 представлены примеры кадров, которые обрабатываются с применением разработанного алгоритма, реализованного на языке python. Данный язык программирования имеет множество библиотек, как для решения задач технического зрения, так и для анализа данных и машинного обучения.

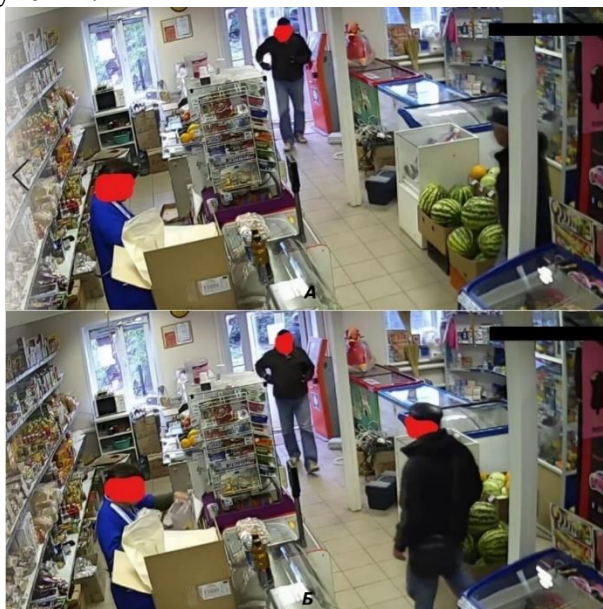


Рис. 2. Пример кадров для обработки

В соответствии с обобщённой блок-схемой алгоритма 2 этап включает обработку изображений. На рис. 3 представлено необработанное (3.А) и обработанное от шумов (3.Б) изображение.

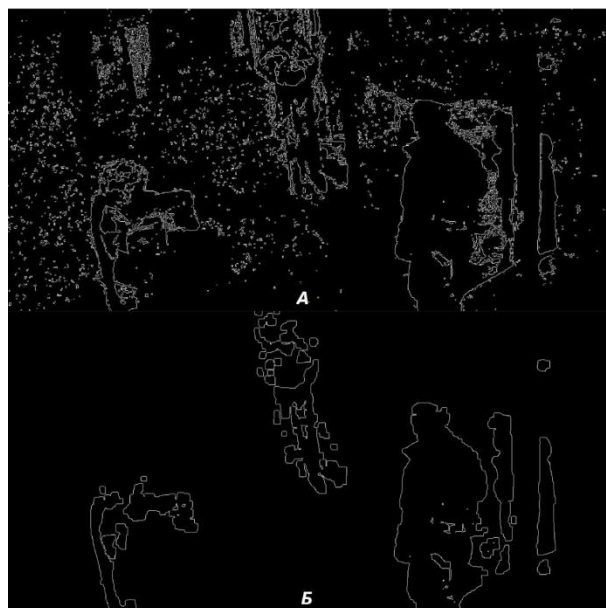


Рис. 3. Пример удаления шумов

Переход от изображения, указанного на рис. 3.А к изображению рис. 3.Б произведен при помощи функции библиотеки OpenCV 'morphologyEx', которая была использована для удаления внешних шумов. За шум были приняты контуры на рис. 3.А, размер которых меньше чем 10×10 пикселей. На рис. 4 представлены результаты обработки изображений.



Рис. 4. Результат обработки изображений

Рис. 4. А и рис. 4. Б представляют два кадра, между которыми происходит поиск разницы изображений. Рис. 4. В иллюстрирует определение объектов по границам их контуров. Стоит заметить, что незначительные части (нижняя часть контура по центру изображения) не выделяются, поскольку алгоритм предусматривает определенные минимальные размеры объекта по высоте. Рис. 4. Г – результат обработки изображений.

Таким образом, в ходе работы алгоритма было замечено, что при низкой динамике объектов алгоритм не находит разницы. Это связано со следующим. При низкой скорости изменения объектом положения в кадре между двумя кадрами есть малые отличия, которые распознаются алгоритмом как шум и устраняются. Для решения данной проблемы предлагается увеличить временной промежуток между двумя сравниваемыми кадрами.

Точность выделения людей изменяющих свои положения в кадре составила не менее 85 процентов. Время выполнения алгоритма составляет 0,058 секунд (среднее время выполнения алгоритма за 1000 вызовов).

Сравнение быстродействия алгоритмов нахождения координат прямоугольников, обрамляющих объекты. Дополнительно был разработан алгоритм, реализация которого имела ту же схему что и на рис. 1, но при реализации нахождения обрамляющих объекты прямоугольников использовалась функция `findContours` библиотеки `OpenCV`. Цель данной функции заключается в определении координаты контуров. После с помощью функции `boundingRect` определялись крайние точки обрамляющих контуры прямоугольников. Затем был написан алгоритм по слиянию пересекающихся прямоугольников. В результате время выполнения всего алгоритма составила 0,061 секунды. От дополнительного алгоритма было принято отказаться, поскольку время выполнения оказалось больше. Существует и другой подход к решению задачи выделения объектов реализованный с помощью каскада Хаара библиотеки `OpenCV`, но время выполнения данного алгоритма дольше на 0.032 секунды, чем работа основного алгоритма, описанного в данной статье.[16]

Аналогичную задачу можно также решить при помощи нейронных сетей, например при обучении с учителем, используя готовую модель из репозитория `TensorFlow object detection API`. [17-20] Время обработки изображения зависит от выбранной предобученной модели.[21] Стоит заметить, что время обработки изображения у некоторых моделей меньше по сравнению временем обработки изображения описанной в данной статье программы. Существенное отличие между такими подходами, как использование нейронных сетей и алгоритма, описанного в данной программе, заключается в том, что нейронные сети, как правило, решают задачу нахождения и классификации объектов в каждом кадре, в то время как описанный алгоритм нацелен на нахождение объекта только при изменении его положения через установленный пользователем промежуток времени.

Заключение. В работе был предложен алгоритм выделения изменяющих свои положения людей в видеопотоке. Для проведения экспериментального исследования была разработана и протестирована программа слежения и выделения движущихся объектов.

В итоге можно сделать следующие выводы:

1. время сравнения двух кадров составляет 0,058 секунд, что означает возможность использования программы в задачах реального времени;
2. точность не менее 85% при съемки внутри помещения.
3. универсальность, которая проявляется в возможности настраивать параметры под разные среды съемки и выделения разных объектов. Например, для фиксирования автомобилей, движущиеся по автомагистрали с превышенной скоростью.

Один из недостатков разработанного алгоритма относительно использования нейронных сетей заключается в настройке параметров алгоритма программы под определенную среду видеосъемки.

Динамический фон изображений также может повлиять на точность выполнения программы из-за увеличения количества и размера шумов.

Данные недостатки не являются критичными при решении задач в помещении или иной среде, где отсутствует или имеется только низкая динамичность заднего фона.

ЛИТЕРАТУРА

Pinnamaneni Bhanu Prasad, Машинное зрение и обработка изображений с применением. (2013). URL:
http://www.researchgate.net/publication/236872855_Machine_Vision_Systems_and_Image_Processing_with_Applications

2. Diogo Carbonera Luvizon, Rodrigo Minetto, Bogdan T Nassu, Измерение скорости машины с помощью обнаружения и отслеживания номерного знака. URL: https://www.researchgate.net/publication/262067228_Vehicle_speed_estimation_by_license_plate_detection_and_tracking.
3. Методы вычитания фона в системе управления технологическим процессом. URL: <https://cyberleninka.ru/article/v/metody-vychitaniya-fona-v-sisteme-upravleniya-tehnologicheskim-protsessom>.
4. Ke Xia, Zhengxin Weng, Система сортировки заготовок на основе промышленного робота машинного зрения.(2016). URL: <https://ieeexplore.ieee.org/document/7810992>
5. К.П. Лошаков, Разработка системы компьютерного зрения для сельскохозяйственной техники.(2018). URL: <http://earchive.tpu.ru/handle/11683/52574>
6. Huan Wu, Huifu Luo, Wei Zhu, Yanghong Wang, Qiang Zhang, Binwu Ma, Yanzhu Yang, Hui Fan, Hongwei Xu, Обнаружение дефектов поверхности гипсового покрытия на основе машинного зрения. (2017). URL: <https://ieeexplore.ieee.org/document/8278354>
7. P Govardhan, Umesh C. Pati, NIR Обнаружение пешеходов на основе изображений в режиме ночного видения с каскадной классификацией и валидацией.(2014). <https://ieeexplore.ieee.org/document/7019339>
8. Dmitry Pertsau, Andrey Uvarov, Алгоритм обнаружения лиц с использованием функции Haar-Like для архитектуры GPU.(2013). <https://ieeexplore.ieee.org/document/6663020>
9. Bradsky G., Kaehler A. Learning OpenCV // O'Reilly, 2008. / С. 18 / ISBN 978-0-596-51613-0.
10. Rezha Aditya Maulana Budiman, Balza Achmad, Faridar, Agus Arif, Nopriadi, Luthfi Zharif, Локализация изображений лейкоцитов с использованием каскадных классификаторов Хаара.(2016) <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7869822>
11. Структура сгенерированного XML-файла: Понимание XML-файла (2017) https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Technical_Guide/050Matching_Records_in_the_Serials_and_Non-Serials_Dedup_Algorithm/030Structure_of_the_XML_File
12. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger (2008). Обзор и оценка часто используемых алгоритмов фонового вычитания (PDF). International Conference on Pattern Recognition. pp. 1–4
13. Zoran Zivkovic, Улучшенная адаптивная модель гауссовой смеси для вычитания фона.(2004) https://scholar.google.com/citations?view_op=view_citation&hl=en&user=GeO26QwAAAAJ&citation_for_view=GeO26QwAAAAJ:u5HHmVD_uO8C
14. Zoran Zivkovic, Ferdinand Van Der Heijden, Эффективная адаптивная оценка плотности на пиксель изображения для задачи вычитания фона.(2006). https://scholar.google.com/citations?view_op=view_citation&hl=en&user=GeO26QwAAAAJ&citation_for_view=GeO26QwAAAAJ:u-x6o8ySG0sC
15. Stefan van der Walt ; S. Chris Colbert ; Gael Varoquaux, The NumPy Array: Структура для эффективного численного расчета. (2011). URL: <https://ieeexplore.ieee.org/document/5725236>
16. Обнаружение человека в реальном времени в компьютерном зрении. URL: <https://medium.com/@madhawavidanapathirana/https-medium-com-madhawavidanapathirana-real-time-human-detection-in-computer-vision-part-1-2acb851f4e55>
17. Васильева Т.Н., Мамонова Т.Е., Применение методов искусственного интеллекта. (2014). URL: <https://elibrary.ru/item.asp?id=23110097>
18. Ashish Sukhadeve, Понимание нейронной сети. (2017). URL: <https://www.datasciencecentral.com/profiles/blogs/understanding-neural-network-a-beginner-s-guide>
19. Elliott Seif, Что такое глубокое обучение? Кто такие учителя глубокого обучения? (2018). URL: <http://inservice.ascd.org/what-is-deep-learning-who-are-the-deep-learning-teachers/>

20. Tensorflow Object Detection API. URL:
https://github.com/tensorflow/models/tree/master/research/object_detection
21. Tensorflow detection model zoo. URL:
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md